**Cognizant - DN 4.0 I Deep Skilling**

**WEEK-5**

**ASP.NET Core 8.0 Web API**

****Lab 6 – Kafka Integration with C#****

**Outline:**

* Introduction to Kafka
* Kafka Architecture
* Topics
* Partitions
* Brokers
* Kafka plug in .NET
* Installation of Kafka
* Basics of Zookeeper
* Demo

**Hands On:**

1. Create a Chat Application which uses Kafka as a streaming platform and consume the chat messages in the command prompt.
2. Create a Chat Application using C# Windows Application using Kafka and consume the message in different client applications.

**SOLUTION :**

**CODE -:**

1. **Producer.cs**

using System;

using System.Threading.Tasks;

using Confluent.Kafka;

public static class Producer // <-- Add 'static' here

{

public static async Task ProduceMessages()

{

var config = new ProducerConfig

{

BootstrapServers = "localhost:9092"

};

using var producer = new ProducerBuilder<Null, string>(config).Build();

Console.WriteLine("Enter messages to send (type 'exit' to quit):");

while (true)

{

var input = Console.ReadLine();

if (input.ToLower() == "exit")

break;

var result = await producer.ProduceAsync("chat-topic", new Message<Null, string> { Value = input });

Console.WriteLine($"✅ Message sent to {result.TopicPartitionOffset}");

}

}

}

1. **Consumer.cs**

using System;

using Confluent.Kafka;

public static class Consumer

{

public static void ConsumeMessages()

{

var config = new ConsumerConfig

{

BootstrapServers = "localhost:9092",

GroupId = "chat-consumer-group",

AutoOffsetReset = AutoOffsetReset.Earliest

};

using var consumer = new ConsumerBuilder<Ignore, string>(config).Build();

consumer.Subscribe("chat-topic");

Console.WriteLine("🔁 Listening for messages (press Ctrl+C to exit)...");

try

{

while (true)

{

var message = consumer.Consume();

Console.WriteLine($"📩 Received: {message.Message.Value}");

}

}

catch (OperationCanceledException)

{

consumer.Close();

}

}

}

1. **Program.cs**

using System;

using System.Threading.Tasks;

class Program

{

static async Task Main(string[] args)

{

Console.WriteLine("Select mode: 1 = Producer, 2 = Consumer");

var choice = Console.ReadLine();

if (choice == "1")

await Producer.ProduceMessages();

else if (choice == "2")

Consumer.ConsumeMessages();

else

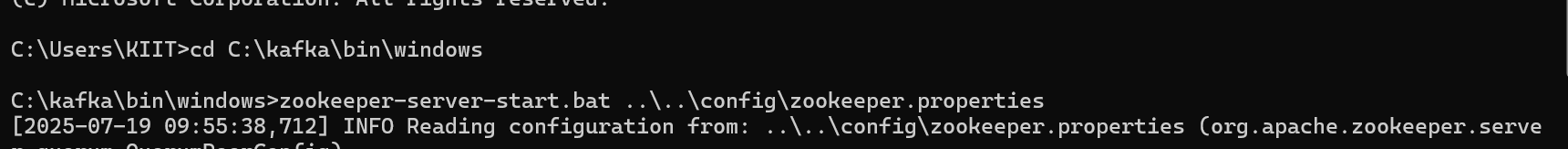
Console.WriteLine("Invalid choice.");

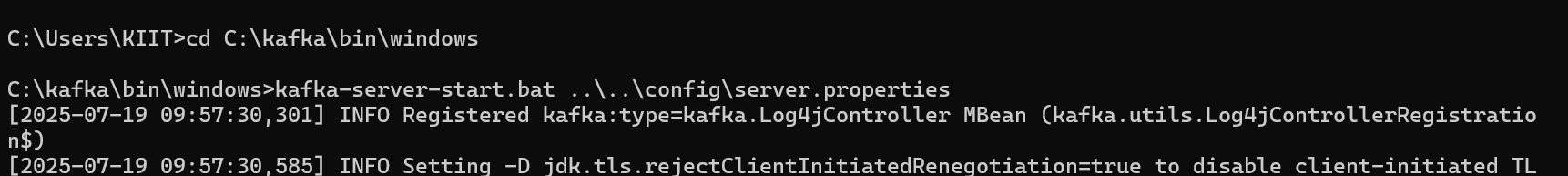
}

}

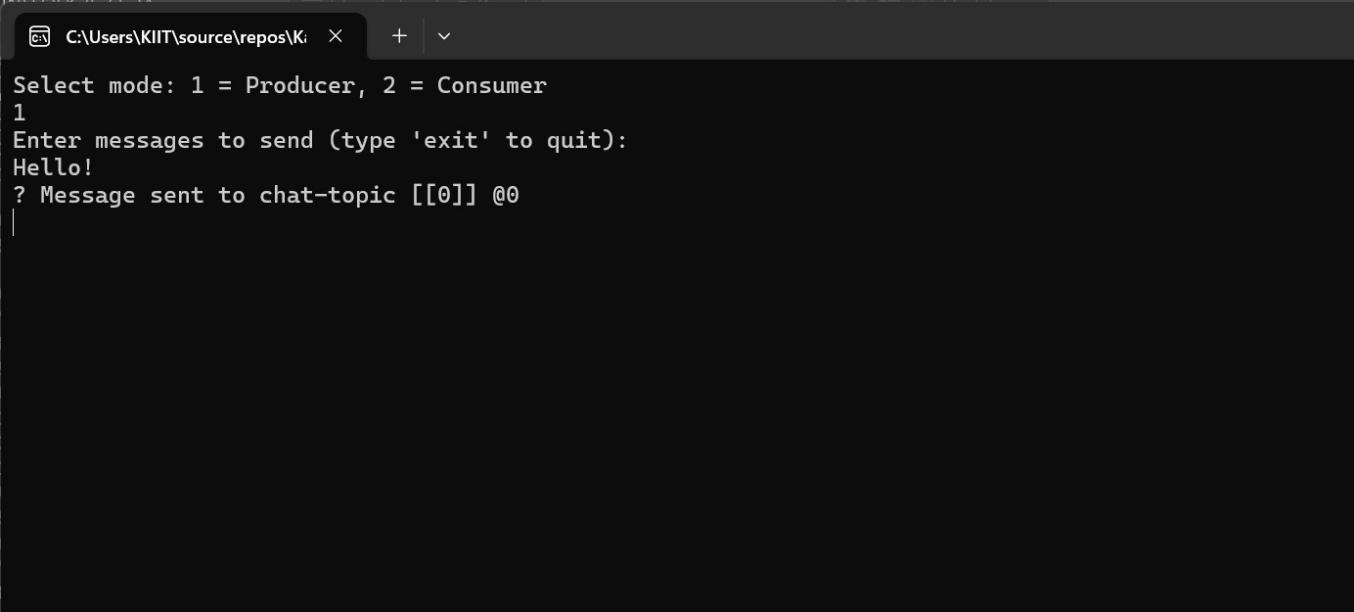
**OUTPUT -:**

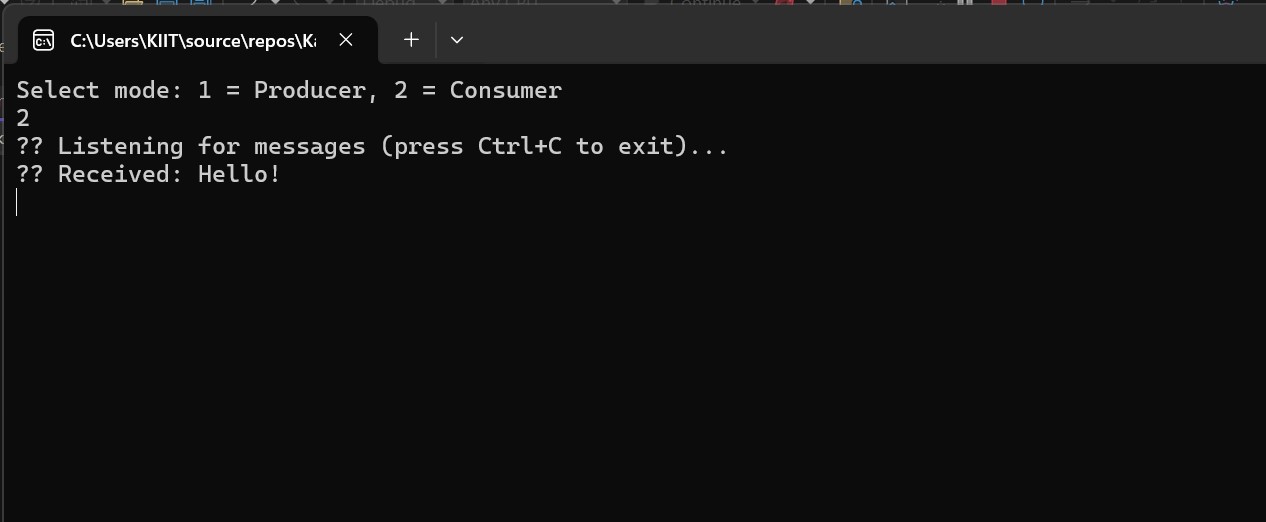
**Starting Zookeeper and Kafka Servers**





**Final Output**

****

****